IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A Review on Fault Detection in Cloud

Shammi Wasim, Dr Jameel Ahmad

#1integral university Lucknow

*2 integral university Lucknow, Associate Professor

***Abstract*— The specifics of this review paper are provided. With the rapid development of cloud technologies and the migration of a lot of cloud-oriented applications to cloud environments, cloud system stability and handling or controlling became increasingly important. Even though cloud monitoring receives less attention, it is essential to perform precise and ongoing monitoring activities in order to accurately operate cloud processes and identify faults. Cloud infrastructure's complex processes can be reviewed, observed, and managed with the aid of cloud monitoring. Multiple entries are added to logs through extended monitoring. Processing these is challenging, as is timely identification of actual flaws and the provision of potential solutions before serious errors occur.**

***Keywords*— cloud technologies**, **cloud monitoring**, **actual faults**, **possible solutions**.

1. INTRODUCTION

Now days with reference to cloud continuous huge data collection with a huge or can say big amount of information is made easier with cloud monitoring. Fault detection is essential for maintaining quick and dependable service. When compared to the total logs gathered from the cloud routine checks, failure data are typically relatively small. That makes it hard to find, evaluate, and fix some of the services' flaws. Here in cloud there are many such type of fault possible like media fault, processor fault, service entry fault, transient, permanent, intermittent fault, network, physical fault, process fault and service entry fault can be listed so that in future improvement done on these category of faults.

According to Farida Asadova et al., the region of the cloud framework in which huge amount of information in file that is called logs are stored and is connected to the categorization of fault in list format [1]. Farooq et al.'s manuscript provides a definition of the general concept of fault [2], which states that "A fault can be defined as abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function." Software executions are linked to failures, which are observable outcomes of faults [2]. Before failures occur, the error that caused them should be identified in order to prevent them. Misinterpretation of the program or incorrect handling during software development are typically the causes of these errors. In this review paper discussed by kumari et al, many different type of fault are categorized as transient, intermittent, permanent, network or software [3]. Cloud computing has come a long way in recent years.

Cloud computing is becoming more prevalent in many internet applications. We can any time able to access application form cloud using internet without downloading these application software these kind of services making cloud special [9]. The cloud facilitates the storage and management of a tremendous amount of data. This data must be protected by cloud computing to prevent corruption and malicious data. Storage, scalability, and reliability are just a few of its features, and it also helps with the management of a lot of data. As a result, cloud computing uses fault tolerance to solve this issue and achieve high performance. Faults cause errors, which in turn cause the system to fail. Cloud failures can only be improved by avoiding failures [9].

A major issue in today's cloud systems is fault recognition and system recovery. It is necessary for a cloud service to be able to quickly identify a problem and address it. Cloud computing is plagued by a wide variety of errors that impair cloud performance. An optimal fault tolerance mechanism is utilized to address this issue and lessen the likelihood of failure. The faults are categorized using long and short-term memory.

In this review paper section one contains the introduction, section two contains the literature review details, section three contains the details about various fault detection technique in cloud, and section four describe the conclusion of this review paper.

2. LITERATURE REVIEW

In cloud systems, fault detection and handling are essential tasks, according to Farida Asadova et al.[1]. Manual interaction and monitoring have become less feasible as these infrastructures grow and change. Monitoring systems are developed to monitor the behavior of cloud system components (such as nodes) and served applications in the virtual environment to address this issue. The majority of cloud environments today offer graphics accelerators to users, resulting in a variety of issues. The use of GPUs in deep learning, on the other hand, may also aid in the identification of inappropriate behavior. In this paper, a short outline of cloud checking and shortcoming discovery strategies is given zeroing in on GPU-empowered hubs.

Mirzaei Davoud and al. [4], A scattered data approximation technique for detecting and approximating the discontinuities of a bivariate function and its gradient is the subject of this paper. Principal component analysis, polyharmonic kernel interpolation, and partition of unity are the foundations of the new algorithm. A set of fault points on or near discontinuity curves is detected using localized polyharmonic interpolation in a partition of unity setting. The detected points are then thinned by moving them roughly along the fault curves using a combination of partition of unity and principal component regression. The fault curves are then reconstructed using a parametric spline interpolation and an ordered subset of these narrowed points. The algorithm's performance is demonstrated by a number of different numerical examples and an application for resolving scalar conservation law equations.

JIE LI and co al., [5] Increasing power equipment's resilience and operational efficacy necessitate real-time smart grid monitoring. For the purpose of continuously monitoring the grid, cloud-based and edge-based fault detection systems that incorporate deep learning have recently been proposed. However, edge-based schemes do not adequately take into account the detection requirement, so they are unable to provide flexible and optimal performance, while state-of-the-art cloud-based detection may necessitate uploading a large amount of data and experience lengthy network delays. We investigate a cloud-edge-based hybrid smart grid fault detection system as a potential solution to these issues. For fault detection, several lightweight neural networks are embedded in devices at the edge of the monitored equipment. An optimal method for allocating communication and computational resources for this cloud-edge based smart grid fault detection system is designed taking into account limited communication resources, the relatively low computation capabilities of edge devices, and the various monitoring accuracies supported by these neural networks. While meeting the requirements for data transmission and processing latency, our method can improve resource utilization and maximize the system's processing throughput. The results of extensive simulations demonstrate that the proposed scheme is superior to comparison schemes. Additionally, we have tested the system's viability and performance in actual situations using a prototype.

Qiyue Li and co. al, [6] Real-time fault detection is crucial to the smart grid's operation. Anomaly detection systems based on deep learning that make use of cloud computing power have become a growing trend in the future. However, due to the significant delay caused by Internet transmission, the delay time between detection and transmission may exceed the limit. Due to the limited computing power of edge devices, the edge-based scheme may not be able to complete all data detection tasks. As a result, we propose a cloud-edge collaborative smart grid fault detection system that is equipped with a lightweight neural network with varying precision for fault detection and is placed next to edge devices. In addition, a deep reinforcement learning-based suboptimal real-time communication and resource allocation strategy is proposed. The solution time can be greatly reduced using this approach, data transmission delays can be avoided, system throughput can be maximized, and communication efficiency can be improved. The simulation results demonstrate that the plan has lower transmission delays and enhances the smart grid detection system's real-time performance.

Yang, Hyunsik, et al. al., [7] A fault detection technique primarily based on metrics is used to guarantee availability in an existing cloud environment. However, as the cloud grows in size and complexity, the current fault detection method makes it difficult to configure the environment. Furthermore, the metric must be accurately understood before it can be utilized. Additionally, whenever the monitoring environment changes, additional adjustments are required. Numerous machine learning-based fault detection and prediction strategies have recently been proposed as a means of resolving these issues. A supervised machine learning method that learns data related to fault situations and, based on this data, detects faults is the machine learning-based fault detection and recovery model that is most frequently proposed in the cloud. Fault learning, on the other hand, is limited by the difficulty of acquiring all of the fault situation data required to learn all of the fault situations that occur in a massive cloud environment. Additionally, when a fault occurs outside of the learned fault pattern, it is challenging to identify it. In addition, it is essential to talk about the automatic recovery architecture that leads to the fault recovery procedure that is based on the results of the fault detection. As a result, we developed and implemented a system as a whole in this paper that uses anomaly detection to anticipate faults.

Lee Yen-Lin, et al. Al,[8] Cloud systems cannot function without effective online liveness fault detection. Cloud system liveness is detected by a single unreliable detector in the majority of current online liveness fault detection methods, such as system layer heartbeating. Regardless of the kind of fault that is detected, a single unreliable detector needs a certain amount of time to detect faults in order to avoid making incorrect judgments. However, other detectors can more quickly identify many faults. As a result, in order to quickly identify faults in cloud systems, this paper proposes an effective online liveness fault detection mechanism that incorporates existing detectors. We compared the proposed mechanism's fault detection effectiveness to that of competing mechanisms. Our proposed mechanism's fault detection time was 70.3% less than that of system layer heartbeating with no additional detection mechanism, according to the findings.

Ashritha Pola, et al. al., [9] In cloud computing, having access to data at any time is important, and one important task is to get data and keep it safe from loss or incursion. A cloud service needs to be able to quickly and accurately respond to unexpected problems. As a result, a fault-finding system employing a variety of methods and algorithms is developed. Cloud computing experiences a variety of faults, which contribute to its subpar performance. Using a long short term memory (LSTM) algorithm and a fuzzy one class support vector machine, the various types of faults that occurred are collected and categorized. Algorithms like the Naive Baye Algorithm, Decision Tree Algorithm, K-Neighbors Algorithm, and Logistic Regression Algorithm are used to compare precision and accuracy. Among the aforementioned algorithms, experimental results demonstrate that logistic regression provides fault detection with the highest accuracy, precision, and performance. The results of our experiments show that our model works.

et al., Denis Sodin al., [10] Every electric power system, whether it is a transmission network (TN) or a distribution network (DN), relies heavily on fault detection and localization to speed up power restoration and improve the system's availability and dependability. A framework for phasor measurement unit (PMU)-based fault detection and localization is described in this paper. The objective was to make the framework viable for DNs, which typically do not have dedicated fiber-optic connectivity available to them, in addition to making the process of fault detecting, localizing, and reporting to the control center completely automated. The evaluation of the quality of service (QoS) for PMU data transmission using the widespread long-term evolution (LTE) technology served as the basis for the proposed edge-cloud framework. The following are a few key benefits of the proposed framework: a) fault detection is performed at the edge nodes, avoiding issues with communication delay and availability; b) potential packet losses are eliminated by temporally storing data at the edge nodes; and c) the amount of data transferred to the control center during steady-state conditions of the network can be significantly reduced because fault detection is no longer centralized but rather takes place locally at the edge.

Hyunsik Yang, et. Al.,[11] The container-based cloud is used in a variety of service infrastructures because it is configurable in both bare-metal and virtual machine (VM) environments and is lighter and more portable than a VM-based infrastructure. The cloud computing infrastructure for the Internet of Things (IoT) is also changing from being based on virtual machines (VMs) to being based on containers. For mission-critical IoT services like real-time health monitoring, vehicle-to-vehicle (V2V) communication, and industrial IoT, the cloud infrastructure's service availability is more important than general computing services in IoT clouds. However, the current fault detection method only takes into account the container's infrastructure in the container environment that runs on a virtual machine, limiting the level of availability required for the operation of mission-critical IoT cloud services. Consequently, fault detection and recovery techniques that take into account both the VM and container levels are required in a container environment running on a virtual machine. We design and implement a Fast Fault Detection Manager (FFDM) architecture for fast fault detection using OpenStack and Kubernetes in this study, which also examines the fault-detection architecture in a container environment. We demonstrated through performance measurements that the FFDM can triple the fault detection time of the previous method.

[YONGJIE ZHANG and others], 12] Because of their complexity and randomness, high impedance faults (HIFs) in distribution networks are difficult to describe and precisely identify. Because of this, data-based methods are a better choice because traditional feature analysis techniques may not be as reliable or as general. However, previous statistical analyses indicate that only a small amount of historical HIF data—less than 20%—can be recorded and utilized in actual situations. Under the cloud-edge collaboration framework of the Internet of Things, a transfer learning-based HIF detection method is proposed in this article to address the issue of outdated data by integrating historical data from multiple distribution networks. A fundamental cloud convolutional neural network model for HIF detection is constructed by first integrating all features from various distribution networks using the cloud-edge collaboration framework. Edge computers use precise synchronous measurements provided by distribution-level phasor measurement units to extract and update the features. During the process of extracting features, principal component analysis is used to bring the data scales of the various distribution networks into a consistent range. By fine-tuning, the target HIF detection model is transferred from the fundamental cloud model to each distribution network in particular. To boost the transferred model's performance, a locality-sensitive hashing-based data augmentation approach is also suggested. Online and offline modes of operation are both possible with the proposed HIF detection method. Seven distinct numerical simulations of distribution networks and one actual experimental distribution network were used to verify the performance.

[Khiet Thanh Bui and others], 13] Both service providers and end users are always concerned about ensuring the availability of cloud computing services. As a result, the system always requires safeguards for unforeseen events. As a result, in order to guarantee both the smoothness and the quality of the services they provide, cloud computing services need to be able to recognize issues and respond appropriately when they do arise. Based on the Fuzzy Oneclass support vector machine and the Exponentially Weighted Moving Average, we propose a fault detection strategy for multi-tier web applications deployed in a cloud computing environment. The feature selection method, which is based on the Random Forest algorithm, is then used to locate the suspicious metrics. Using TPC-W (TPC BenchmarkTM W, which simulates the activities of a business-oriented transaction web server in a controlled internet commerce environment), a multi-tier application is deployed by a transnational web e-Commerce benchmark in the private cloud and injected with typical faults to evaluate our strategy. The results of the experiment show that the fault detection and diagnosis work well.

3. Various Fault Detection in Cloud Systems

Even though cloud systems are getting bigger and better, system failures will always happen. The cloud system's fault detection requires the quick processing of monitoring data to accurately predict system failures and avoid potential obstacles. Failure is not the result of every fault; the severity of each fault should be prioritized. The term "failure" describes the idea of "reliability," which in turn describes the level of service provided by cloud service providers. Based on the article by Smara et al., the following methods are utilized for cloud system fault detection: 14]. Intrusion or anomaly detection system is the first one [14].

The host or network intrusions are the primary focus of this kind of detection. Behavior analysis is used to find anomalies. The subgroups of intrusion detection are signature-based and anomaly-based detection. A predetermined database of previously experienced anomalies with particular priority is included in signature-based detection. However, the anomaly-based detection method searches the log data for unusual patterns. In particular, the article by Abbasi et al. presents an efficient anomaly detection system for fault-tolerant network management in cloud data centers [15]. Software-defined networking makes administration simpler, has features for controlling the network, and has a programmable console. This feature, which can be scheduled for the reading network by subnets, aids in the utilization of the best network path and cloud network fault management [15].

Anomaly-based detection techniques based on statistics, data mining, and machine learning are available in cloud computing. The statistical type of detection [14] is able to identify unpredictable flaws by comparing the data to the stored ideal conditions. In contrast, the data mining detection method [14] uses rule-based technologies like Classification, Clustering, and Association to identify flaws. This method can tell the difference between actions that are typical and wrong. Because it doesn't require any prior knowledge to process the data, it's a popular method. The primary drawback of this method is that it may result in numerous false alerts. The system stores the values of the errors it finds using the machine learning detection method. Through its capacity to learn from the faults' previous values, the system enhances its performance. While machine learning is capable of fault detection on its own, it is significantly more effective when combined with statistical and data mining techniques. As a result, hybrid approaches are being utilized in cloud systems, which only incur computational costs.

The heartbeat and pinging method is the second strategy [14]. The heartbeat method uses a monitoring device that constantly checks the fault detector to see if the fault is present. The heartbeat method considers the device to be incorrect if the detector does not respond before timeout. In the pinging method, the fault detector confirms the malfunction by sending a message to the monitoring device. For persistent hardware fault detection, both approaches are utilized. Analyzing the collected data is yet another approach to handling it. It is impossible to rely on a human supervisor to identify all issues in such a system with extensive data; some automation is essential. Organizations suffer significant financial losses as a result of server outages. For instance, during a one-hour interruption on Prime Day, Amazon lost $72 million [16].

Additionally, insufficient memory issues and service interruptions in cloud systems are brought on by collective remote access to virtual machines. Legashev et al.'s paper says [17] An efficient scheduling strategy employing Simulated Annealing and Genetic Algorithm is presented after collective access issues in educational institutions are investigated. Cloud monitoring as a service continuously monitors the applications' behavior and measures the infrastructure [18]. A data-gathering module, a data-processing module, and user and application interfaces make up this kind of tool [19]. The environment's messages are continuously gathered and stored during data collection. A visualization tool may be part of data processing to help explain the current state.

Zhang and others 20] discovered that cloud fault detection falls into two broad categories: rule-based and statistically based detection. Depending on the error message and record components, basic decision trees can be constructed using multiple rules and queries or simple rulesets for rule-based detection methods. The similarity of previously discovered flaws is based on other methods. Using the so-called macrostep-based execution, Lovas [21] proposed a method and a framework that not only actively controls cloud-based deployment processes but also monitors them. Manual rule-based evaluation of each macrostep, or collective breakpoint set, is made possible by the method. Machine learning is now used in statistics-based methods [20]: Based on input observations, neural networks (NNs) and SVMs are frequently utilized as classificators [24–27]. In Zuzana et al.'s article, [28] In the process of detecting dysphonia, statistically based functionalities are created to extract features from sample speech data.

4. CONCLUSION

We used a variety of sources and documentation to focus on the broad overview in this review. Here in cloud there are many such type of fault possible like media fault, processor fault, service entry fault, transient, permanent, intermittent fault, network, physical fault, process fault and service entry fault can be listed so that in future improvement done on these category of faults. This review paper major focus on to give a overview about many different type of fault which is occurred in cloud. The prepared overview opens up brand-new research avenues and has the potential to make significant contributions to the study of orchestrated cloud application debugging.

REFERENCES

[1] Farida Asadova, G´abor Kert´eszy, R´obert Lovasz and S´andor Sz´en´asi, “Fault detection in GPU-enabled Cloud Systems -An Overview”, IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics • March 2-5, 2022.

[2] S. U. Farooq, S. Quadri, and N. Ahmad, “Metrics, models and measurements in software reliability,” in 2012 IEEE 10th international symposium on applied machine intelligence and informatics (SAMI). IEEE, 2012, pp. 441–449.

[3] P. Kumari and P. Kaur, “A survey of fault tolerance in cloud computing,” Journal of King Saud University-Computer and Information Sciences, vol. 33, no. 10, pp. 1159–1176, 2021.

[4] Davoud Mirzaei, Navid Soodbakhsh,” A fault detection method based on partition of unity and kernel approximation”, Numerical Algorithms https://doi.org/10.1007/s11075-022-01488-4, 2023

[5] JIE LI, RUIDONG LI, ZHI LIU, “Resource Orchestration of Cloud-edge based Smart Grid Fault Detection”, Association for Computing Machinery. 1550-4859/2022.

[6] Qiyue Li, Yadong Zhu, Weitao Li,” Deep Reinforcement Learning based Resource Allocation for Cloud Edge Collaboration Fault Detection in Smart Grid”, DOI: 10.17775/CSEEJPES.2022.02390, CSEE Journal of Power and Energy Systems, 2022

[7] Hyunsik Yang, Younghan Kim, “Design and Implementation of Machine Learning-Based Fault Prediction System in Cloud Infrastructure”, https://doi.org/10.3390/electronics11223765, Electronics 2022, 11, 3765.

[8] Yen-Lin Lee, Deron Liang, Wei-Jen Wang, “Optimal Online Liveness Fault Detection for Multilayer Cloud Computing Systems”, DOI 10.1109/TDSC.2021.3100680, IEEE Transactions on Dependable and Secure Computing, 2021.

[9] Pola Ashritha, M Banusri, R Namitha, Dr J Shiny Duela,” Effective fault detection approach for cloud computing”, Journal of Physics: Conference Series 1979 (2021) 012061 doi:10.1088/1742-6596/1979/1/012061, 2021.

[10] Denis Sodin, Urban Rudež, Marko Mihelin, Miha Smolnikar,” Advanced Edge-Cloud Computing Framework for Automated PMU-Based Fault Localization in Distribution Networks”, Appl. Sci. 2021, 11, 3100. https://doi.org/10.3390/app11073100 https://www.mdpi.com/journal/applsci, 2021.

[11] Hyunsik Yang and Younghan Kim,” Design and Implementation of Fast Fault Detection in Cloud Infrastructure for Containerized IoT Services”, Sensors 2020, 20, 4592; doi:10.3390/s20164592 www.mdpi.com/journal/sensors, 2020.

[12] YONGJIE ZHANG, XIAOJUN WANG, YIN XU,” A Transfer Learning-Based High Impedance Fault Detection Method Under a Cloud-Edge Collaboration Framework”, date of publication September 8, 2020, date of current version September 22, 2020. *Digital Object Identifier 10.1109/ACCESS.2020.3022639. 2020.*

[13] Khiet Thanh Bui, Len Van Vo, Canh Minh Nguyen, Tran Vu Pham, and Hung Cong Tran,” A Fault Detection and Diagnosis Approach for Multi-tier Application in Cloud Computing”, JOURNAL OF COMMUNICATIONS AND NETWORKS, VOL. 22, NO. 5, OCTOBER 2020.

[14] M. Smara, M. Aliouat, A.-S. K. Pathan, and Z. Aliouat, “Acceptance test for fault detection in component-based cloud computing and systems,” Future Generation Computer Systems, vol. 70, pp. 74–93, 2017.

[15] A. Abbasi, S. Band, M. A. A. Al-qaness, A. Abbasi, N. Al-Jallad, and A. Mosavi, “Resource-aware network topology management framework,” Acta Polytechnica Hungarica, 02 2020.

[16] F. A. April Berthene, Josh Neblett, “Digital commerce 360. the potential cost of amazon’s prime day,” https://www.digitalcommerce360.com/ 2018/07/17/the-potential-cost-of-amazons-prime-day-miss-72-million/,
Aug 2018, [Online] Accessed 9 October 2021.

[17] L. V. Legashev and I. P. Bolodurina, “An effective scheduling method in the cloud system of collective access, for virtual working environments,” Acta Polytechnica Hungarica, vol. 17, pp. 179–195, 2020.

[18] G. Aceto, A. Botta, W. De Donato, and A. Pescap`e, “Cloud monitoring: A survey,” Computer Networks, vol. 57, no. 9, pp. 2093–2115, 2013.

[19] J. Montes, A. S´anchez, B. Memishi, M. S. P´erez, and G. Antoniu, “Gmone: A complete approach to cloud monitoring,” Future Generation Computer Systems, vol. 29, no. 8, pp. 2026–2040, 2013.

[20] P. Zhang, S. Shu, and M. Zhou, “An online fault detection model and strategies based on svm-grid in clouds,” IEEE/CAA Journal of Automatica Sinica, vol. 5, no. 2, pp. 445–456, 2018.

[21] B. Ligetfalvi, M. Em˝odi, J. Kov´acs, and R. Lovas, “Fundamentals of a novel debugging mechanism for orchestrated cloud infrastructures with macrosteps and active control,” Electronics, vol. 10, no. 24, 2021.

[22] P. Kacsuk, “Systematic macrostep debugging of message passing parallel programs,” Future Generation Computer Systems, vol. 16, no. 6, pp. 609–624, 2000.

[23] Y. Tamura, Y. Nobukawa, and S. Yamada, “A method of reliability assessment based on neural network and fault data clustering for cloud with big data,” in 2015 2nd International Conference on Information Science and Security (ICISS). IEEE, 2015, pp. 1–4.

[24] T. Wang, W. Zhang, J. Wei, and H. Zhong, “Fault detection for cloud computing systems with correlation analysis,” in 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM). IEEE, 2015, pp. 652–658.

[25] Q. Liu, F. Zhang, M. Liu, and W. Shen, “A fault prediction method based on modified genetic algorithm using bp neural network algorithm,” in 2016 IEEE International Conference on Systems, Man, and Cybernetics
(SMC). IEEE, 2016, pp. 004 614–004 619.

[26] P. Zhang, S. Shu, and M. Zhou, “Adaptive and dynamic adjustment of fault detection cycles in cloud computing,” IEEE Transactions on Industrial Informatics, vol. 17, no. 1, pp. 20–30, 2019.

[27] X. Zhang and Y. Zhuang, “A fault detection algorithm for cloud computing using qpso-based weighted one-class support vector machine,” in International Conference on Algorithms and Architectures for Parallel Processing. Springer, 2019, pp. 286–304.

[28] Z. Dankoviˇcov´a, D. Sov´ak, P. Drot´ar, and L. Vokorokos, “Machine learning approach to dysphonia detection,” Applied Sciences, vol. 8, no. 10, p. 1927, 2018.